

Handwritten initials: "AF" with a flourish above it.

Appeal Brief Under 37 C.F.R. § 41.37
Attorney Docket No.: 019287-0319645
Application Serial No.: 10/759,705



IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

APPELLANTS :	Billy KEEFER et al.	CONFIRMATION No.:	8442
SERIAL NUMBER :	10/759,705	EXAMINER:	Karen C. Tang
FILING DATE :	January 15, 2004	ART UNIT:	2151
FOR : SYSTEM AND METHOD FOR AGENT-BASED MONITORING OF NETWORK DEVICES			

**Appellants' Brief on Appeal
Under 37 C.F.R. § 41.37**

Mail Stop Appeal Brief - Patents

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Sir:

Further to the Notice of Appeal dated **July 3, 2007**, Appellants hereby submit this Appellants' Brief of Appeal pursuant to 37 C.F.R. § 41.37.

The Director is authorized to charge the fee for filing an Appeal Brief pursuant to 37 C.F.R. § 41.20(b)(2), as well as any additional fees that may be due, or credit any overpayment of same, to Deposit Account No. 033975 (**Ref. No. 019287-0319645**).

Appeal Brief Under 37 C.F.R. § 41.37

I. Real Party in Interest

By virtue of the assignment recorded at Reel 015448, Frame 0621, Computer Associates Think, Inc., the assignee of the present application, is the real party in interest.

II. Related Appeals and Interferences

Appellants are not aware of any related appeals or interferences.

III. Status of Claims

Pending: Claims 1-27 are pending.
Cancelled: No claims have been cancelled.
Rejected: Claims 1-27 stand rejected.
Allowed: No claims have been allowed.
On Appeal: Claims 1-27 are appealed.

IV. Status of Amendments

No amendments to the claims have been filed subsequent to the Final Office Action dated April 3, 2007 (hereinafter "Final Action").

V. Summary of Claimed Subject Matter

The following exemplary citations to the Specification and/or drawing figures are not exclusive, as other examples of support for claimed subject matter exist. As such, the following citations should not be viewed as limiting.

Independent Claim 1

According to various aspects of the invention, as recited in claim 1, for example, a method for agent-based monitoring of network devices in an enterprise network may comprise selecting one of the network devices, each of which have characteristics associated therewith, from the enterprise network (e.g., Specification at 5, lines 18-25).

Subsequently, to monitor hardware characteristics of the network device, one of a plurality of agent templates may be selected based on one or more of the characteristics of the selected network device (e.g., Specification at 7, line 29 – 8, line 3). For example, the agent template may include a hierarchy of object classes, each of which corresponds to a possible combination of the characteristics of the selected network device (e.g., Specification at 8, lines 14-19). Accordingly, an agent object may be instantiated from the object class of the agent template corresponding to the characteristics of the selected network device (e.g., Specification at 8, lines 20-26) in order to monitor hardware characteristics of the network device (e.g., Specification at 8, line 26 – 9, line 14).

Independent Claim 9

According to various aspects of the invention, as recited in claim 9, for example, software stored on a machine-readable medium may include executable instructions for agent-based monitoring of network devices in an enterprise network (e.g., Specification at 3, lines 2-7). For example, agent-based monitoring of network devices may include selecting one of the network devices, each of which have characteristics associated therewith, from the enterprise network (e.g., Specification at 5, lines 18-25).

Subsequently, to monitor hardware characteristics of the network device, one of a plurality of agent templates may be selected based on one or more of the characteristics of the selected network device (e.g., Specification at 7, line 29 – 8, line 3). For example, the agent template may include a hierarchy of object classes, each of which corresponds to a possible combination of the characteristics of the selected network device (e.g., Specification at 8, lines 14-19). Accordingly, an agent object may be instantiated from the object class of the agent template corresponding to the characteristics of the selected network device (e.g., Specification at 8, lines 20-26) in order to monitor hardware characteristics of the network device (e.g., Specification at 8, line 26 – 9, line 14).

Independent Claim 17

According to various aspects of the invention, as recited in claim 17, for example, a system for agent-based monitoring of network devices in an enterprise network a memory

operable to store information associated with a plurality of network devices in the enterprise network, where the information stored in the memory includes characteristics of each of the plurality of network devices (e.g., Specification at 5, lines 18-25).

Subsequently, to monitor hardware characteristics of the network device, one or more processors may collectively select a plurality of agent templates based on one or more of the characteristics of the selected network device (e.g., Specification at 7, line 29 – 8, line 3). For example, the agent template may include a hierarchy of object classes, each of which corresponds to a possible combination of the characteristics of the selected network device (e.g., Specification at 8, lines 14-19). Accordingly, the processors may instantiate an agent object from the object class of the agent template corresponding to the characteristics of the selected network device (e.g., Specification at 8, lines 20-26) in order to monitor hardware characteristics of the network device (e.g., Specification at 8, line 26 – 9, line 14).

Independent Claim 25

According to various aspects of the invention, as recited in claim 25, for example, a method for agent-based monitoring of switches in an enterprise network may comprise selecting one of the switches, which has characteristics associated therewith, from the enterprise network (e.g., Specification at 5, lines 18-25).

Subsequently, to monitor hardware characteristics of the selected switch, one of a plurality of agent templates may be selected based on one or more of the characteristics of the selected switch (e.g., Specification at 7, line 29 – 8, line 3). For example, the agent template may include a hierarchy of object classes, each of which corresponds to a possible combination of the characteristics of the selected switch (e.g., Specification at 8, lines 14-19).

Accordingly, an agent object may be instantiated from the object class of the agent template corresponding to the characteristics of the selected switch (e.g., Specification at 8, lines 20-26) in order to monitor hardware characteristics of the selected switch by comparing at least one of the hardware characteristics to a threshold value (e.g., Specification at 8, line 26 – 9, line 14; and 16, lines 4-13). Furthermore, when at least one of the hardware

characteristics violates the threshold value, an alert may automatically be communicated in response thereto (e.g., Specification at 16, lines 12-13).

VI. Grounds of Rejection to be Reviewed on Appeal

Claims 1-27 stand rejected under 35 U.S.C. § 103(a) as allegedly being obvious over U.S. Patent No. 6,816,898 to Scarpelli et al. ("Scarpelli") in view of U.S. Patent No. 6,842,781 to Lavian et al. ("Lavian"). Final Action at 3-10.

VII. Argument

The Examiner has rejected claims 1-27 under 35 U.S.C. § 103(a) as allegedly being obvious over Scarpelli in view of Lavian. Final Action at 3-10. This rejection is improper, and must be reversed, for at least the reason that the Examiner has failed to establish a *prima facie* case of obviousness, as the references relied upon, either alone or in combination, do not disclose, teach, or suggest every feature of the claimed invention. For at least this reason, the rejection is improper and must be reversed.

More particularly, Scarpelli and Lavian, either alone or in combination, fail to disclose, teach, or suggest at least the feature of "selecting one of a plurality of agent templates based on one or more of the characteristics of the selected network device, the agent template comprising a hierarchy of object classes, wherein each object class corresponds to a possible combination of the characteristics of the selected network device," as recited in claim 1, for example. The Examiner acknowledges the deficiency of Scarpelli with respect to at least this feature of the claimed invention. Final Action at 3. Nonetheless, the Examiner alleges that Lavian teaches selecting an agent template "based on one or more of the characteristics of the selected network device," and further alleges that Lavian teaches an agent template "comprising a hierarchy of object classes, . . . wherein each object class corresponds to a possible combination of the characteristics of the selected network device." Final Action at 4. Appellants disagree with the Examiner's assessment.

For example, the passages of Lavian relied upon by the Examiner specifically relate to a management information database (MIB) map for "converting object-oriented requests for

MIB information into requests for network parameters.” Lavian at col. 5, lines 36-39. In this regard, the Examiner equates a request for “parameters from the network devices” with “selecting one of a plurality of agent templates based on one or more of the characteristics of the selected network device,” as recited in claim 1. However, contrary to the Examiner’s assertions, Lavian does not disclose, teach, or suggest selecting an agent template based on characteristics of a network device, as recited in claim 1.

Rather, Lavian expressly describes querying the MIB map to request method calls that can access or manipulate network parameters. To this end, the MIB map includes “a database listing the network parameters related to the management of a network device and a set of object-oriented methods for manipulating the network parameters.” Lavian at col. 5, lines 45-49. Accordingly, Lavian relates, at best, to a system that queries the same database (i.e., the MIB map) in response to any and all “requests . . . to access and manipulate the network parameters” associated with a device. Lavian at col. 5, lines 48-51. By contrast, claim 1 recites, among other things, “**selecting** one of **a plurality** of agent templates based on one or more of the characteristics of the selected network device.” Lavian clearly does not disclose, teach, or suggest the MIB map being selectively queried, as all requests are routed through the MIB map. Nor does Lavian disclose, teach, or suggest the MIB map being “one of a plurality of agent templates,” as the MIB map does not serve as an agent template because it includes the information associated with all network devices sought to be managed.

In fact, other passages of Lavian explicitly describe “a framework for executing a variety of mobile agents,” but notably these passages do not disclose, teach, or suggest the MIB map or the information contained therein being part of the agent framework. Further still, Lavian does not disclose, teach, or suggest that the agent framework includes “a plurality of agent templates based on one or more of the characteristics of the selected network device,” as recited in claim 1, for example. Rather, Lavian expressly states that “a mobile agent [can] operate on a variety of network devices and operating environments.” Lavian at col. 6, lines 31-27. As a result, none of the agents described in Lavian can operate as an “agent template comprising a hierarchy of object classes, wherein each object class corresponds to a possible combination of the characteristics of the selected network device.” By being generic to “a

variety of network devices and operating environments,” Lavian does not disclose, teach, or suggest selecting an agent template that includes “a hierarchy of object classes, wherein each object class corresponds to a possible combination of the characteristics of the selected network device,” as recited in claim 1, for example.

In other words, the aforementioned passages of Lavian unequivocally state that any given agent can interface with various different execution environments or various different kinds of network devices. Thus, even assuming *arguendo* that the mobile agents described in Lavian can be considered agent templates, Lavian expressly precludes each object class or other program structure associated with the agents from necessarily corresponding “to a possible combination of the characteristics of the selected network device.” Instead, Lavian allows other network devices or execution environments to be associated with an agent, which contradicts various features recited in claim 1.

Furthermore, Lavian does not disclose, teach, or suggest that the MIB map comprises “a hierarchy of object classes, wherein each object class corresponds to a possible combination of the characteristics of the selected network device,” as recited in claim 1, for example. For example, the passages of Lavian relied upon by the Examiner as allegedly teaching this feature unambiguously state that the MIB map receives “requests for network parameters,” which are mapped to “a set of object-oriented **methods** for manipulating the network parameters.” Lavian at col. 5, lines 40-55 (emphasis added). Thus, the MIB map described by Lavian includes, at best, associations between a given network parameter and a **method** for manipulating the parameter.

By contrast, claim 1 recites, among other things, an agent template that includes “a hierarchy of **object classes**, wherein each object class corresponds to a possible **combination** of the characteristics of the selected network device.” The passages of Lavian relied upon by the Examiner, however, clearly do not disclose, teach, or suggest a hierarchical organization of object classes, as object-oriented methods are technically distinct from object classes for various apparent reasons. Furthermore, the passages of Lavian clearly indicate that each object-oriented method included in the MIB map corresponds to a **method** for manipulating a **singular**, independent network parameter. Thus, Lavian clearly does not disclose, teach, or

suggest “selecting one of a plurality of agent templates,” wherein the selected agent template includes “a hierarchy of object classes,” each of which “corresponds to a possible combination of the characteristics of the selected network device,” as recited in claim 1, for example. For at least these reasons, the references relied upon, either alone or in combination, do not disclose teach, or suggest every feature of claim 1.

Accordingly, for at least the foregoing reasons, the Examiner has failed to establish a *prima facie* case of obviousness. In particular, neither Scarpelli nor Lavian, either alone or in combination, disclose, teach, or suggest at least the feature of “selecting one of a plurality of agent templates based on one or more of the characteristics of the selected network device, the agent template comprising a hierarchy of object classes, wherein each object class corresponds to a possible combination of the characteristics of the selected network device,” as recited in claim 1, for example. For at least this reason, the rejection is improper and must be reversed.

Moreover, neither Scarpelli nor Lavian, either alone or in combination, disclose, teach, or suggest at least the feature of “instantiating an agent object from the object class of the agent template that corresponds to the characteristics of the selected network device,” as recited in claim 1, for example. The Examiner alleges that Scarpelli teaches this feature at col. 7, lines 14-27, and at Figures 6a and 7. Final Action at 3. Appellants disagree with the Examiner’s assessment.

Scarpelli specifically relates to incorporating user-generated script programs into a network monitor, which “collects meta data and data defined by the script-based programs.” Scarpelli at col. 7, lines 14-27. However, the execution of a script, as described by Scarpelli, is distinct from “instantiating an agent object” for at least the reason that scripts (e.g., the shell scripts discussed in Scarpelli) are not object-oriented program structures. Rather, scripts have various characteristics that distinguish them from “an agent object [that can be instantiated] from the object class of [an] agent template,” as recited in claim 1. For example, Scarpelli acknowledges various distinctions between scripts and object-oriented programs by specifically indicating that certain devices or applications can be represented as objects. However, the Examiner specifically relies upon the script-based programs as allegedly teaching the

aforementioned feature of the claimed invention, but script-based programs are not objects that can be instantiated from an object class, as recited in claim 1, for example.

Accordingly, for at least the reason that Scarpelli relates to scripts that are not instantiated objects (or object classes associated with a template from which objects can be instantiated), Scarpelli fails to disclose, teach, or suggest “instantiating an agent object from the object class of the agent template,” as recited in claim 1, for example.

Appellants further note that similar arguments as presented herein have been raised previously, and the Examiner responded to the arguments by alleging “that the features upon which applicant relies (i.e., Object-oriented program structure [*sic*]) are not recited in the . . . claim(s).” Final Action at 2. Appellants take exception to the Examiner’s allegation that the claims do not recite object-oriented program structures. For example, independent claim 1 explicitly recites that “agent templates” comprise “a hierarchy of ***object classes***,” and that an “***instantiated agent object***” results from “instantiating an ***agent object*** from the ***object class*** of the agent template.” Thus, the Examiner’s allegation that object-oriented program structures are not recited in the claims is blatantly false. For at least these reasons, the Examiner’s arguments in this regard carry no weight.

Furthermore, also in response to the previous arguments, the Examiner alleges that Scarpelli discloses a meta registration process that is used to create/instantiate agent objects, in which “the meta is a [*sic*] agent template . . . comprising object/device information, which is [the] object class.” Final Action at 2. These arguments of the Examiner clearly establish how the Examiner is misinterpreting the claimed invention, as well as the references alleged as teaching the features of the claimed invention. Specifically, Scarpelli states that the “registration process allows the administrator to define the . . . new service monitor,” which “can then be activated to monitor ***any applicable devices, applications, or servers in the network***.” Scarpelli at col. 8, lines 47-54. As a result, Scarpelli simply does not disclose, teach, or suggest that the meta registration process is “one of a plurality of agent templates.” The Examiner attempts to equate a registration process (e.g., execution of a series of instructions) with an agent template that includes “a hierarchy of object classes.” As such, the rejection is

clearly improper for at least the reason that the Examiner alleges technically distinct things to be equivalents.

Moreover, even if the meta registration process could be considered an agent template (which Appellants note is not the case), Scarpelli does not disclose, teach, or suggest the process utilizing “a hierarchy of object classes, wherein each object class corresponds to a possible combination of the characteristics of the selected network device.” On the contrary, Scarpelli specifically indicates that the registration process collects data defined by script-based programs, and that the registration process uses the script-based program in a monitor that can “monitor any applicable devices, applications, or servers in the network.” Scarpelli at col. 7, lines 28-33; col. 8, lines 47-54. As a result, for at least the reason that Scarpelli collects object/device information using a script, the meta registration process is unrelated to an “agent template comprising a hierarchy of object classes.”

For similar reasons, a device that is monitored by Scarpelli is distinct from an “object class [that] corresponds to a possible combination of the selected network device.” In other words, an “object class corresponds to . . . the selected network device,” such that the object class is distinct from the device itself. The Examiner, however, improperly equates one with the other. Furthermore, because Scarpelli specifically describes a singular monitor that can “monitor any applicable devices, applications, or servers in the network,” Scarpelli clearly does not disclose, teach, or suggest “a hierarchy of object classes, wherein each object class corresponds to a possible combination of the characteristics of the selected network device.” That is, even if the registration process were considered analogous to an “agent template,” as alleged by the Examiner, the monitor created thereby could “be activated to monitor any applicable devices, applications, or servers in the network.” Thus, at best, the monitor would be able to invoke a plurality of scripts, but each script would not necessarily correspond “to a possible combination of the characteristics of the selected network device.” As such, the Examiner’s alleges that the references apply against the claimed invention in a way that specifically contradicts at least the feature in claim 1 reciting that “each object class corresponds to a possible combination of the characteristics of the selected network device.”

Accordingly, for at least the foregoing reasons, the Examiner has failed to establish a *prima facie* case of obviousness. In particular, neither Scarpelli nor Lavian, either alone or in combination, disclose, teach, or suggest at least the feature of “instantiating an agent object from the object class of the agent template that corresponds to the characteristics of the selected network device,” as recited in claim 1, for example. For at least this reason, the rejection is improper and must be reversed.

Claims 9, 17, and 25 include features similar to those set forth in claim 1. Claims 2-8, 10-16, 18-24, and 26-27 depend from and add features to one of claims 1, 9, 17, and 25. Thus, the rejection of these claims is likewise improper and must be withdrawn for at least the same reasons.

VIII. Claims Appendix

The pending claims (claims 1-27) are attached in **Appendix A**.

IX. Evidence Appendix

Appendix B: None.

X. Related Proceedings Appendix

Appendix C: None

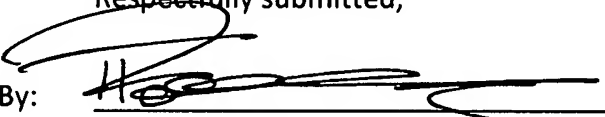
Conclusion

For at least the foregoing reasons, Appellants respectfully submit that the claims are clear, definite, and allowable over the references relied upon by the Examiner. Therefore, reversal of the rejections is respectfully requested.

Date: **September 4, 2007**

Respectfully submitted,

By:


Rick A. Toering
Registration No. 43,195

PILLSBURY WINTHROP SHAW PITTMAN LLP
P.O. Box 10500
McLean, Virginia 22102
Main: 703-770-7900
Fax: 703-770-7901

Appendix A: Claims Appendix

1. **(Previously Presented)** A method for agent-based monitoring of network devices in an enterprise network, comprising:
 - selecting one of the network devices from the enterprise network, each network device having characteristics;
 - selecting one of a plurality of agent templates based on one or more of the characteristics of the selected network device, the agent template comprising a hierarchy of object classes, wherein each object class corresponds to a possible combination of the characteristics of the selected network device; and
 - instantiating an agent object from the object class of the agent template that corresponds to the characteristics of the selected network device, the instantiated agent object operable to monitor hardware characteristics of the network device.
2. **(Previously Presented)** The method of claim 1, wherein the characteristics of the network device include at least one Management Information Base (MIB) parameter.
3. **(Previously Presented)** The method of claim 1, wherein the characteristics include one or more of a type of network device, an identity of a vendor, a model number, a product line, or a hardware characteristic.
4. **(Previously Presented)** The method of claim 1, wherein monitoring includes retrieving information associated with one or more of the hardware characteristics of the network device.
5. **(Previously Presented)** The method of claim 4, wherein the hardware characteristics of the network device include one or more of:
 - memory usage;
 - chassis temperature;

Central Processing Unit (CPU) usage;
fan status;
module status; or
power supply status.

6. **(Previously Presented)** The method of claim 4, wherein monitoring includes comparing a threshold value to the retrieved information associated with one or more of the hardware characteristics.

7. **(Previously Presented)** The method of claim 6, further comprising automatically communicating an alert in response to the hardware characteristic violating the threshold value.

8. **(Previously Presented)** The method of claim 1, wherein the hierarchy of object classes includes a plurality of parent objects and at least one child object associated with each of the parent objects, the parent objects corresponding to different embodiments of a first characteristic of the network device and each child object being associated with different embodiments of a second characteristic and the embodiment of the first characteristic that corresponds to the parent object associated with the child object.

9. **(Previously Presented)** Software comprising executable instructions stored on a machine-readable medium, the software operable to:

select one of the network devices from the enterprise network, each network device having characteristics;

select one of a plurality of agent templates based on one or more of the characteristics of the selected network device, the agent template comprising a hierarchy of object classes, wherein each object class corresponds to a possible combination of the characteristics of the selected network device; and

instantiate an agent object from the object class of the agent template that corresponds to characteristics of the selected network device, the instantiated agent object operable to monitor hardware characteristics of the network device.

10. **(Previously Presented)** The software of claim 9, wherein the characteristics of the network device include at least one MIB parameter.

11. **(Previously Presented)** The software of claim 9, wherein the characteristics include one or more of a type of network device, an identity of a vendor, a model number, a product line, or a hardware characteristic.

12. **(Previously Presented)** The software of claim 9, wherein monitoring hardware characteristics includes retrieving information associated with one or more of the hardware characteristics of the network device.

13. **(Previously Presented)** The software of claim 12, wherein the hardware characteristics of the network device includes one or more of:

- memory usage;
- chassis temperature;
- Central Processing Unit (CPU) usage;
- fan status;
- module status; or
- power supply status.

14. **(Previously Presented)** The software of claim 12, wherein monitoring hardware characteristics includes comparing a threshold value with at least one of the hardware characteristics.

15. **(Previously Presented)** The software of claim 14, further operable to automatically communicate an alert in response to the at least one of the hardware characteristics violating the threshold value.

16. **(Previously Presented)** The software of claim 9, wherein the agent object includes a parent object and at least one child object, the parent object associated with the network device and each child associated with one of the hardware characteristics.

17. **(Previously Presented)** A system for agent-based monitoring of network devices in an enterprise network, comprising:

- memory operable to store information associated with a plurality of network devices in the enterprise network, the information stored in the memory comprising characteristics of each of the plurality of network devices; and

- one or more processors collectively operable to:

- select one of the network devices from the enterprise network;

- select one of a plurality of agent templates based on one or more of the characteristics of the selected network device, the agent template comprising a hierarchy of object classes, wherein each object class corresponds to a possible combination of the characteristics of the selected network device; and

- instantiate an agent object from the object class of the agent template that corresponds to the characteristics of the selected network device, the instantiated agent object operable to monitor hardware characteristics of the network device.

18. **(Previously Presented)** The system of claim 17, wherein the characteristics of the network device include at least one MIB parameter.

19. **(Previously Presented)** The system of claim 17, wherein the characteristics include one or more of a type of network device, an identity of a vendor, a model number, a product line, or a hardware characteristic.

20. **(Previously Presented)** The system of claim 17, wherein the instantiated agent object includes processors operable to retrieve information associated with one or more of the hardware characteristics of the network device.

21. **(Previously Presented)** The system of claim 20, wherein the hardware characteristics of the network device include one or more of:

- memory usage;
- chassis temperature;
- Central Processing Unit (CPU) usage;
- fan status;
- module status; or
- power supply status.

22. **(Previously Presented)** The system of claim 20, wherein the agent object compares a threshold value to the retrieved information associated with one or more of the hardware characteristics.

23. **(Previously Presented)** The system of claim 22, wherein the agent object automatically communicates an alert in response to one or more of the hardware characteristics violating the threshold value.

24. **(Previously Presented)** The system of claim 17, wherein the hierarchy of object classes includes a plurality parent objects and at least one child object associated with each of the parent objects, the parent objects corresponding to different embodiments of a first characteristic of the network device and each child object being associated with different embodiments of a second characteristic that corresponds to the parent object associated with the child object.

25. **(Previously Presented)** A method for agent-based monitoring of switches in an enterprise network, comprising:

selecting one of the switches from the enterprise network, each switch having characteristics;

selecting one of a plurality of agent templates based on one or more of the characteristics of the selected switch, the agent template comprising a hierarchy of object classes, wherein each object class corresponds to a possible combination of the characteristics of the selected network device; and

instantiating an agent object from the object class of the agent template that corresponds to the characteristics of the selected network device, the instantiated agent object operable to monitor hardware characteristics of the selected switch by comparing at least one of the hardware characteristics to a threshold value, and

automatically communicating an alert in response to the at least one of the hardware characteristics violating the threshold value.

26. **(Previously Presented)** The software of claim 9, the characteristics comprising one or more of:

- a device type;
- a device vendor;
- a hardware characteristic;
- a model number; or
- a product line.

27. **(Previously Presented)** The software of claim 9, the software further operable to:

transmit using Simple Management Network Protocol (SNMP) a request for a Management Information Base (MIB) object from the selected network device, wherein the MIB object identifies a type of the network device; and

identify a class table containing a plurality of agent templates wherein the one of the plurality of agent templates is selected from the class table based on the type of the network device.

Appendix B: Evidence Appendix

None.

Appendix C: Related Proceedings Appendix

None.